

REMARKS

In the Office Action, the Examiner rejected Claims 1-13, which were all of the then pending claims, under 35 U.S.C. §103 as being unpatentable over U.S. patent application no. 2002/0049833 (Kikinis) in view of U.S. Patent 6,262,729 (Marcos, et al.) and a Microsoft Press "Computer Dictionary," 3rd Edition (Microsoft Dictionary). It is noted that the previous rejection of the claims under 35 U.S.C. §103 as being unpatentable over Kikinis and the Microsoft Dictionary has been withdrawn.

Applicant herein asks that independent Claims 1, 6 and 10 be amended to better define the subject matters of these claims. Also, Claims 14-16, which are dependent from Claims 1, 6 and 10 respectively, are being added to describe preferred features of the invention.

For the reasons discussed below, Claims 1-16 patentably distinguish over the prior art and are allowable. The Examiner is, thus, asked to reconsider and to withdraw the rejection of Claims 1-13 under 35 U.S.C. §103, and to allow these claims and new Claims 14-16.

The present invention, generally, relates to processing text files used to communicate between applications or between an application and an end user. As explained in the present application, there are three significant problems in text file processing. A first problem is how to describe the program for the structure of the text file, a second problem is mapping data between the text file and the application, and a third problem is how to describe the flow of control needed to process the file.

This invention effectively addresses all three of these problems. The first problem is solved by forming the templates so that they have literal fragments of the text file. These templates are then used as overlays for parsing the text file into segments, or as prototypes to generate segments of output files. The second problem – mapping data from the text file to

the application -is solved by the use of specialized macro classes. For example, input macros are provided to read in a segment of the text file and to use that segment to initiate application update processing.

The third problem - flow control - is addressed by the interaction of templates and macros. A macro - or, more precisely, a pointer to the template - is stored as a keyword in the template. During the parsing of the text file, when that pointer is reached, the pointer is used to invoke the macro, and the macro is used to map data from one of the segments of the text file to the computer application. The macro, in turn, may call for another template to further parse the text file, and this second template may call other macros.

The present invention distinguishes over the prior art not because the present invention uses templates and macros, but because of the way in which the templates and macros are used and the way in which the templates and macros work together to control the processing flow.

It is important to emphasize that the present invention is about processing text files, not about producing HTML. The Kikinis and Marcos patents are specifically limited to HTML or some other means of Web presentation. Also, although the present application uses the words "templates" and "macro," these features are used in this invention in specific ways, described in the application.

The primary reference, Kikinis, relied on to reject the claims uses the word "template," but it is not clear what this "template" is. In Kikinis, there is no reference to storing the template as text, and it appears unlikely that a set of parameters would be stored in simple text format.

More particularly, paragraphs 13 and 14 of Kikinis refer to "creating a list of parameters" and "storing the parameters as a template" (in paragraph 13) and then (in paragraph 14) "The template comprises one or more parameters derived from the characteristics of the client device." It should be quite evident to one with even a minimal familiarity with the art that what is described here is very different from the templates of the present invention. The templates of the present invention comprise text fragment with places to "fill in the blanks." What Kikinis is describing is nothing of the sort.

With particular regard to Claim 2, it is noted that Kikinis refers to "script." Presumably, however, the references to a script in Kikinis (paragraphs 17-22), are references to a "Mark-Script." The only similarity between this feature of Kikinis and the present invention is that both have some means to invoke computer processing. Since, as mentioned above, the templates described in Kikinis are not the same as the templates of the present invention, then there is no possibility that an interaction with the templates of Kikinis via a script could be the same as the interaction of templates and macros that happens in the preferred embodiment of the invention. Beyond that, Kikinis does not describe a specific interaction between the Mark-Scripts and the templates of that invention.

Marcos (column 9, lines 5-15) talks about the initial template definition being stored in text format, but not about support for editing fragments in text format. The editing features described in Marcos are fancier and specifically limited to adding HTML definitions. In addition, in Marcos, the entire HTML page is stored in one chunk, rather than being composed of separate, independently edited fragments.

The Microsoft Dictionary was cited for its definition of several terms, including "literals." This Dictionary, though, does not address the problems that are solved by the present invention, and thus this reference does not provide any guidance or suggestion as to how to modify the prior art, including Kikinis, to solve those problems effectively.

Independent Claims 1, 6 and 10 are being amended to describe more clearly the above-discussed feature of the invention. In particular, Claims 1 and 10 are being amended to describe the steps of forming a template having literal fragments of the text file, providing a macro class to map data from the text file to the computer application, and embedding in the template a pointer to the macro class. Claims 1 and 6 describe the further step of using the template as an overlay to parse the text file into segments having data, or as a prototype to generate a segment of an output file; and these claims describe this using step as including the steps of reaching said pointer in the template; and when that pointer is reached, using the pointer to invoke the macro class, and using the macro class to map data from one of the segments of the text file to the computer application.

Claim 6, which is directed to a system for processing a text file in a computer application is being amended to describe means forming a template having literal fragments of the text file, means forming a macro class to map data from the text file to the computer application, wherein a pointer to the macro class is embedded in the template. Claim 6 also sets forth means for using the template as an overlay to parse the text file into segments having data; and Claim 6 describes this using means as including means for using the pointer, when that pointer is reached in the template, to invoke the macro class, and to use the macro class to map data from one of the segments of the text file to the computer application.

The prior art of record fails to disclose or suggest the above-discussed combination of templates and macros, and the way in which the templates and macros function together to process a text file. Kikinis discloses templates, but the templates are not used to parse the text file into segments. Kikinis also discloses scripts, but these scripts are not used to map data from one of the segments of the text file, as segmented by the template, to the computer application.

The other references of record have been reviewed, and they too, whether they are considered individually or in combination, do not disclose or suggest using a template/macro class combination in the manner described in Claims 1, 6 and 10.


For example, as previously mentioned, the Microsoft Dictionary was cited for its definition of several terms, including "literals." This reference, however, does not address the same problems that are addressed by the present invention. Hence, this Dictionary does not provide any specific suggestion as to how to modify the prior art provide to solve those problems effectively.

Because of the substantial differences between Claims 1, 6 and 10 and the prior art, and because of the advantages associated with those differences, it cannot be said that any of these claims is obvious to one of ordinary skill in the are. Accordingly, these claims patentably distinguish over the prior art and are allowable. Claims 2-5 and 14 are dependent from, and are allowable with, Claim 1; and Claims 7-9 and 15 are dependent from Claim 6 and are allowable therewith. Likewise, Claims 11-13 and 16 are dependent from Claim 10 and are allowable therewith.

It is noted that the changes requested herein to Claims 1, 6 and 10 are being made in response to the suggestion from the Examiner in the Office Action to amend the claims in this way. In addition, applicant's previous Amendment overcame the previous rejection of the claims, and this Amendment is the first opportunity Applicant has had to address the latest rejection of the claims over the combination of Kikinis and the Microsoft Dictionary. It is believed that Applicant should have an opportunity to respond to that rejection without having to file a continuation or an RCE. Accordingly, it is believed that entry of this Amendment is appropriate, and such entry is respectfully requested.

For the reasons set forth above, the Examiner is asked to enter this Amendment, to reconsider and to withdraw the rejection of Claims 1-13 under 35 U.S.C. §103, and to allow these claims. If the Examiner believes that a telephone conference with Applicant's Attorneys would be advantageous to the disposition of this case, the Examiner is asked to telephone the undersigned.

Respectfully submitted,


John S. Sensny
Registration No. 28,757
Attorney for Applicant

Scully, Scott, Murphy & Presser
400 Garden City Plaza
Garden City, New York 11530
(516) 742-4343

JSS:jy